Proposing an architecture for developing tactical shooter games employing Virtual Reality

Ícaro S. Barbosa[1], Creto A. Vidal[1], Joaquim Bento C. Neto[1], Antônio José M. L. Júnior[2] George Allan M. Gomes[2]

[1]Departamento de Computação – Universidade Federal do Ceará (UFC)
Ceará, Brazil

[2]Instituto Virtual – Universidade Federal do Ceará (UFC)
Ceará, Brazil

icrsbarbosa@gmail.com, {cvidal,joaquimb}@lia.ufc.br,

{melojr,george}@virtual.ufc.br

Abstract. *Tactical Shooters are a trendy subgenre within FPS games, focusing on realistic simulations of armed combat. With the advancement of Virtual Reality (VR) technology, which is becoming increasingly prevalent in household settings. In this article, we present an architecture that addresses the development of Tactical Shooters using Virtual Reality, aiming to provide players with enhanced immersion. To validate the proposed model, we developed two prototypes in which we analyzed the time and difficulty in developing them.*

1. Introduction

Competitions involving shooting have been taking place ever since projectile weapons existed. However, we can pinpoint the famous carnivals of the 1890s as the starting point for the development of first-person shooter (FPS) games [Wolf 2012]. Over time, electromechanical devices replaced the tent stands and eliminated the need for a human operator to reset the targets after each round.

Over the years, the development and utilization of various other game interaction devices, such as mouse and keyboard, gesture capture [Amir et al. 2016], or special controllers like the Wii Remote [Júnior et al. 2012], have facilitated the emergence of branches within the FPS genre, giving rise to subgenres such as Shooting Gallery, Rail Shooters, and notably, Tactical Shooters [Adams 2014].

Tactical Shooters are a subgenre of FPS games focusing primarily on simulating modern combat. They aim to provide realistic weaponry and situations, offering a gameplay experience that resembles the reality of handling weapons and making decisions in an actual combat scenario [Adams 2014].

The recent advancements in immersive Virtual Reality (VR) and the decreasing cost of Head-Mounted Displays (HMDs) made those devices increasingly prevalent in domestic settings. They are considered now as a promising technology for simulators in the fields of security and defense [de Armas et al. 2020], and consequently, in Tactical Shooters as well. Therefore, in this article, we propose an architecture for developing immersive VR-based Tactical Shooters, aiming to encourage the growth of this genre of games on these new platforms.

The remainder of this paper is organized in two sections. In Section II, we describe the methodology, including the identified requirements, the proposed architecture and its elements, and the prototypes already implemented based on the architecture. Finally, in Section III, we present our final remarks and provide some suggestions for future work, as well as the expected contributions of this article.

2. Methodology

In this section, we define the model, the proposed architecture, and its foundation. We also showcase the prototypes developed.

2.1. Requirement

Essential elements of FPS games are defined in [Adams 2014], and these include:

- The nature of the interactions associated with the player's objective to win the game.
- The nature of the targets – their types and behaviors when they are hit or not.
- The nature of the player's representation (avatar) – its behaviors, and its interaction possibilities with the game's elements (environment, weapons, targets etc.).

In FPS Tactical Shooters, some of these essential elements can also be found in studies that have developed models for shooting simulators for police training. While these works initially define simulation requirements, it is noteworthy that Tactical Shooters can be considered simulators that closely resemble games [Tekinbas and Zimmerman 2003]. The developers of the game/simulator America's Army stated in [Davis 2004]: "We are well positioned for the future of defense modeling and simulation. And that future has a game face."

Based on the initial analysis of the Virtra V-300 [v30], a commercial shooting simulator that uses a CAVE system, Júnior et al. [Júnior et al. 2012] identified essential requirements for this type of simulation, including:

- Simulation of specific situations to force the trainee to make quick and proper decisions, including the execution of shots or making alternative decisions, respecting the doctrine previously established.
- Communication, management and control solutions based on modular system (computer, screen, speaker and shooting capture system).
- Use of specific systems for measuring a trainee's performance, considering variables such as response time to events and shot positions.

We can reuse and modify those requirements, adapting them to the best use in the specific game context without compromising the training. Thus, we can define a general requirement for the development of FPS Tactical Shooters, which must necessarily consider the following:

- Well-defined environments with direct responses to the player's decisions.
- Well-established avatar behaviors involving all possibilities of interaction with the environment.
- Well-defined behaviors of the targets present in the environment.
- Good management of the necessary visual and sound responses to represent the various occurrences in the game.
- Clearly defined objectives with ways to measure progress, such as score calculation and overall performance of the player.

2.2. Architecture

As a way of enabling the creation of such Tactical Shooters, in conformity with the previously defined requirements, we propose the architecture presented in Figure 1, which consists of the following elements:

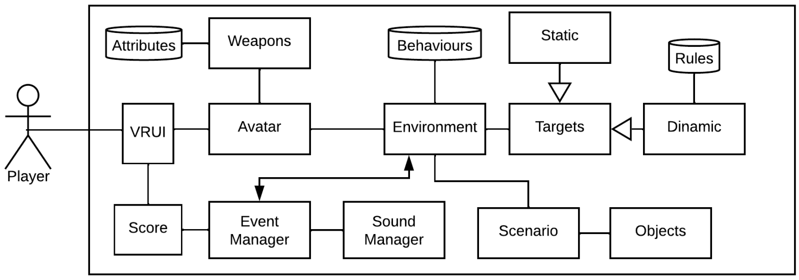

Figura 1. Proposed architecture.

- The **Player** is the user who will play the FPS game using immersive virtual reality equipment such as HMDs.
- The **VRGUI** is the interface the Player uses to interact with the game through visual and auditory stimuli and controls from the virtual reality system, such as the controls typically associated with HMDs.
- The interactions performed by the Player, through controls and their movements in the physical environment, mapped to the virtual environment, will be replicated by the **Avatar**, representing the Player in the FPS environment.
- The Avatar will be able to interact with **Weapons**, which may have **Attributes** that simulate the behavior and characteristics of real weapons, such as accuracy, range, and ammunition capacity, for example, [Adams 2014]. For example, **Attributes** can be used as controls for the amount of ammunition and restrictions on periods when the Player can or cannot shoot.
- The **Scenario** will be the Scene composed of Objects in which the Avatar can move and interact. The Objects present in the Scenario will simulate natural static objects, which can represent supports for the Targets, for example, or be used as barriers and shelters, allowing the Player to create his specific game strategies.
- **Targets** are entities that can be Static or Dynamic. Static targets do not have movement and are always active, meaning shots from Weapons can hit them. Dynamic targets have their own **Rules** for activation, movement, and deactivation. Examples of **Rules** in the Targets include disappearing when hit, appearing as criminal or innocent, and moving along predefined paths.
- The **Environment** will consist of a combination of the Scenario, Avatar, and Targets. It will have its own **Behaviours**, allowing the interaction of each of the three entities with each other and potentially modifying the overall gameplay, managing the Attributes and Rules so that the game behaves in the necessary way that is intended to be created in the Scenario. Some **Behaviors** examples include managing the number of shots fired, spawning targets at specific times, and keeping track of scores.

- The **Event Manager** will capture all events generated by both the Avatar and the Targets and Objects in the Scene, which will generate auditory feedback by Sound Manager, if necessary, and calculate the respective Score that can be stored and later displayed to the Player through the VRGUI.

2.3. Prototypes

Two game prototypes were developed based on the established architecture: All-Range, and Closed Range. These prototypes were chosen due they are activities used in shooting trainings.

The prototypes were created using the Unity 3D game engine [uni] and compiled for the Oculus Quest 2 HMD (Head-Mounted Display) [ocu]. The Unity engine was chosen for its popularity and extensive documentation, primarily supported by its active community.

The Raycast algorithm implemented in Unity was utilized to determine the positions where the projectiles fired by the weapons collide with the targets. To create the prototypes, the model was developed by individually implementing each of the components presented in the architecture and was then configured according to the needs of each prototype. The following are the descriptions of the two defined prototypes:

2.3.1. Closed Range

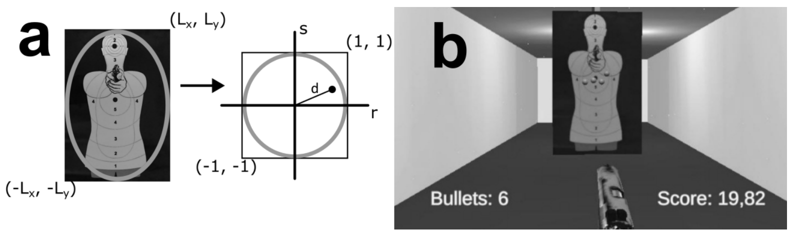

Figura 2. a) Floor plan of the Closed Range prototype. The square is where the player can move, and the blue rectangle is where the target moves. b) UI view of the Closed Range prototype. c) The scoring calculation for the Closed Range prototype.

Adopting the proposed architecture for developmente, this prototype allows the player to use one of the controllers to handle the weapon and fire shots. For each shot, a score is calculated based on how far the hit position is from the target's chest center. In this prototype, one gets 5 points for hitting the target's chest center and a smaller and smaller value the further away from the center the hit is. The calculation performed is $5(1 - min(d, 1))$, where $d = \sqrt{r^2 + s^2}$, $r = \frac{x}{L_x}$, and $s = \frac{y}{L_y}$ how is shown in Figure 2.a. The corresponding score is updated and presented at the bottom right corner. The number of shots fired before the target's distance changes is displayed at the bottom left corner (see Figure 2.b). As always, the information in the HUD remains visible during the game. After completing 30 shots, the target moves closer to the player, so he can review the final result with the positions of the hits.

Using the proposed architecture, this prototype was created in a total of 20 hours, including developing all algorithms needed.

2.3.2. All-Range

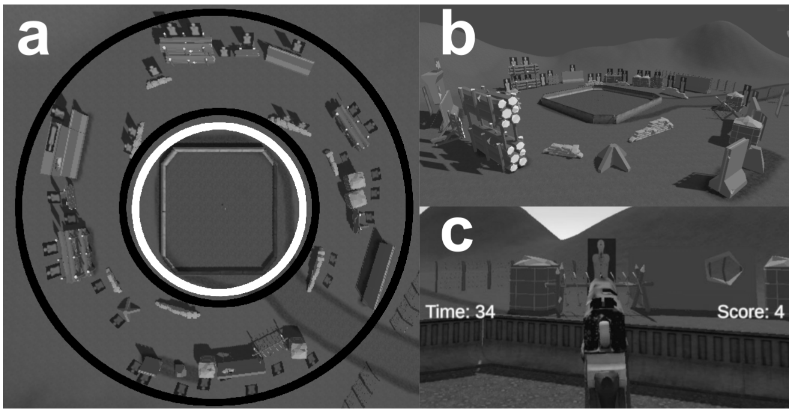

Figura 3. a) Floor plan of the All-Range prototype. The player can move inside the white circle, and the targets appear inside the black torus. b) Front view of the All-Range prototype. c) UI view of the All-Range.

The all-range prototype (Figures 3) consists of a virtual open area delimited by barricades, where the player can freely move. The barriers and targets surround the entire game region, forming a 360° space around the player's avatar. In this prototype, a random target appears at a predefined position one at a time. The player must locate where the target is solely based on the sound it emits upon arising. The target disappears, emitting a different sound, after being hit or after an elapsed time of 10 seconds. Then, another target appears at a random position. The score is calculated based on the number of targets hit within 1 minute, with the increase of 1 point per hit. The remaining time and the score are displayed at the bottom of the screen, in the left and right corners, respectively, as shown in Figure 3.c. During the game, the score and the remaining time stay on the HUD.

This prototype was developed using as a base the previous one. It was developed in around 12 hours since modifying the original Scenario significantly was necessary. However, the developer managed to reuse the Target Rules, thus speeding up development.

3. Final considerations

This study proposed an architecture model for FPS games, specifically Tactical Shooters, employing VR resources. Two prototypes were implemented based on this architecture to validate the proposal.

Based on the prototypes developed, knowing that the structure of the games will follow the same as theirs, and based on the fact that the construction of the prototypes was greatly simplified, it is inferred that the construction of other titles must also be facilitated. It is also worth noting that the developer felt comfortable using the architecture, following the established components and their relationships, as seen in the time and effort required to develop the prototypes shown here.

In future work, we want to conduct tests to more comprehensively validate our proposal, inviting other developers to create their games. We also intend to develop more prototypes to evaluate the extensibility capacities of the proposed architecture.

Referências

Meta quest 2: Immersive all-in-one vr headset — meta store. https://www.meta.com/quest/products/quest-2/. Accessed: 2023-05-24.

Unity real-time development platform — 3d, 2d, vr ar engine. https://unity.com/. Accessed: 2023-05-24.

Virtra v-300® firearms training simulator for law enforcement — virtra. https://www.virtra.com/simulator/law-enforcement-v-300/?lang=us-en. Accessed: 2023-05-24.

Adams, E. (2014). *Fundamentals of Shooter Game Design*. New Riders.

Amir, M. H., Quek, A., Sulaiman, N. R. B., and See, J. (2016). Duke: enhancing virtual reality based fps game with full-body interactions. In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*, pages 1–6.

Davis, M. (2004). *America's Army PC Game Vision and Realization*. US Army and the Moves Institue.

de Armas, C., Tori, R., and Netto, A. V. (2020). Use of virtual reality simulators for training programs in the areas of security and defense: a systematic review. *Multimedia Tools and Applications*, pages 3495–3515.

Júnior, A. J. M. L., Gomes, G. A. M., Junior, N. A. C., dos Santos, A. D., Vidal, C. A., Cavalcante-Neto, J. B., and Gattass, M. (2012). System model for shooting training based on interactive video, three-dimensional computer graphics and laser ray capture. In *2012 14th Symposium on Virtual and Augmented Reality*, pages 254–260. IEEE.

Tekinbas, K. S. and Zimmerman, E. (2003). *Rules of play: Game design fundamentals*. MIT press.

Wolf, M. J. P. (2012). *Battlezone and the Origins of First-Person Shooting Games*. Approaches to Digital Game Studies. Bloomsbury Publishing.